INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(54) Title: APPARATUS FOR PATTERN RECOGNITION
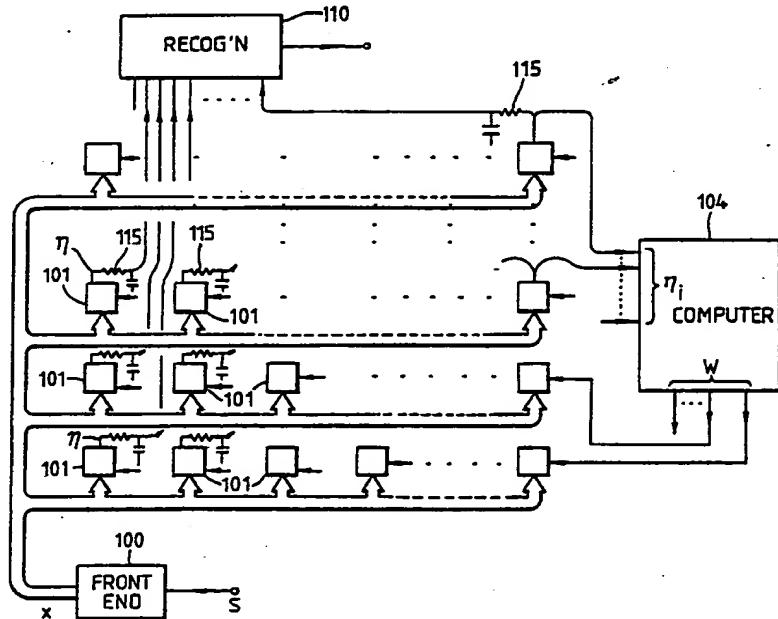
(57) Abstract

A temporal sequence of signal vectors X (e.g. generated from a speech signal) is supplied in parallel to elements (101) in an array each of which (after training of the array) provide an output η indicating the vector's similarity to a reference vector. Persistence is (115) built into the elements so that signals forming a "trajectory" within the array, corresponding to an input sequence are simultaneously available for recognition (unit 110 or, preferably by a further such array). Arrays may be cascaded to allow for longer sequences.

APPARATUS FOR PATTERN RECOGNITION

The invention relates to methods and apparatus for pattern recognition, and particularly, though not exclusively, speech recognition.

5   The invention is particularly concerned with monitoring temporal sequences of input vectors so as to recognise particular patterns.

In this specification, an "N dimensional vector" comprises a group of N values, each value corresponding to 10   a respective dimension of the vector. The values may be represented by analogue signals or digitally. Such a vector has a magnitude which may be, for example, defined by the square root of the sum of the squares of the values, and also a direction in N dimensional space. For 15   simplicity, throughout this specification, scalar quantities (except for N) will be represented by lower case letters, and vector quantities by upper case letters.

Vectors of this type can in particular be derived from an analysis of human speech. Thus, an analogue signal 20   representing a series of speech sounds can be regularly sampled and the content of each sample can be represented in terms of a vector comprising a set of feature values corresponding, for example, to the amplitude of respective frequencies within the sample.

25   A paper entitled "Clustering, Taxonomy, and Topological Maps of Patterns" by T Kohonen in Proceedings of the Sixth International Conference on Pattern Recognition, October 1982, pages 114-128 describes an approach for the statistical representation of empirical 30   data. Sets (vectors) of input data are successively applied, in parallel to each of a number of processing

units regarded as forming a two-dimensional array; each unit produces a single output proportional to the degree of matching between the particular input vector and an internal vector associated with that unit. An adaptation principle is defined so that a succession of input vectors, which form a statistical representation of the input data, cause changes in the internal vectors. This works (for each input vector) by:

(1) identifying the unit whose reference vector is most similar to the input (eg the smallest Euclidean distance);

(2) defining a neighbourhood within the array, around this unit;

(3) changing the internal vectors of those units belonging to this neighbourhood; the direction of change being such tat the similarity of those internal vectors is increased.

As this 'self-organisation' process proceeds, the size of the neighbourhood is progressively reduced; the magnitude of the adjustments may also decrease. At the conclusion of this process, the array internal vectors define a mapping of the input vector space onto the two-dimensional space. Kohonen trained such an array using manually-selected speech samples of certain stationary Finnish vowel phonemes (selected to exclude those including transients), the input vectors each consisting of fifteen spectral values, and found that it mapped the phonemes into the two-dimensional array space.

According to one aspect of the present invention, there is provided a pattern recognition apparatus comprising:

- an input for receiving a temporal sequence of input signal vectors;

- store means for storing a plurality of reference vectors;

- a like plurality of comparison elements arranged in operation to receive the same, current, value of the input vector and to generate an output signal indicative of the degree of similarity between the input vector and a respective one of the reference vectors;

- means for producing for each element an output modified in dependence on the output signal produced by that element as a result of a comparison between its reference vector and at least the immediately preceding input vector; and

- recognition means for comparing the pattern represented by those of the said modified outputs which indicate a relatively high similarity between the input and reference vectors with reference information to identify patterns represented by the said temporal sequence.

This enables the path through the "most similar reference vectors" corresponding to a series of input vectors to be relatively easily determined by building in a certain degree of persistence in the output signals generated for each reference vector. Thus, at any sampling period instant, although the most similar reference vector may have changed, output signals of a significant magnitude will also be generated as a result of the comparison of a preceding input vector with at least one of the other reference vectors.

Although in principle the recognition means could employ a conventional pattern recognition method to determine from the output signals generated an identity for the stimulus leading to the series of input vectors, it is preferred that the apparatus further comprises at least one further array of such comparison elements having respective stored reference vectors, each of which elements is connected to receive a respective group of the

modified (or further modified) outputs of the or a preceding array so as to generate an output signal indicative of the similarity between its reference vector and the said group of outputs.

Preferably, the group of output signals corresponding to each reference vector of the, or one of the, higher arrays is derived from the output signals generated by locations in the preceding array centred on a location in the preceding array corresponding to the location of the reference vector in the one array.

In a further refinement of the multilayer method, the output signals generated by at least one of the arrays are fed back so as to modify the groups of signals fed to the preceding array. This type of feedback enables the method to react to high order grammars (in the case of speech recognition such as words, phrases and the like). For example, the fed back signals may be concatenated with the input signals to the preceding array.

There is a considerable complexity in the subsequent recognition of a series of input vectors if the position within the array which is associated with the most similar reference vector changes with the magnitude of the input vector. This causes a considerable problem in the case of speech recognition since the magnitude of the vector is associated with the intensity of the received sound. In other words, if the same phoneme is said loudly or softly, this will result in different "most similar reference vectors" being located. This in turn means that a complex pattern recognition system must be involved to cope with a wide variety of different input vectors being generated for the same phoneme.

We have found that a considerable improvement on the known methods can be achieved by comparing the input vectors and the reference vectors in such a way that the

position within the array of the reference vector which is most similar to the input vector does not vary with the magnitude of the input vector. This means that, in the case of speech, regardless of the loudness of the sound, substantially the same reference vector will be found to be the most similar.

In the preferred method, the comparison step comprises determining a function related to the dot product of the input vector and the reference vector. In this case, one (preferably the reference vector) or both of the reference and input vectors is normalised.

This has the consequence that the largest excitation in the array comes from the element whose reference (or weight) vector lies most closely in the same direction as the input vector so that the <u>relative</u> excitation depends solely on sound type and not amplitude. However, the absolute magnitude of the excitation will depend on the magnitude of the input vector X, and so this arrangement gives the desired property of coding sound type by position of maximum excitation and sound magnitude by the intensity of that excitation.

In a modified form of the dot product function, the comparison step comprises determining the function:

$$|X| \cos^m \theta$$

where m = a power greater than unity, X is the magnitude of the input vector, and θ is the angle between the input and reference vectors. In this case, the reference vectors have unity magnitude.

Although the apparatus has been described conceptually as involving topological arrays of reference vectors, in practice the physical locations of reference

vectors need not be in a topological manner but will be mapped in a conventional manner into a topological form.

The term "arrays" will be understood in terms of the connections between elements rather than their physical location.

Some examples of apparatus in accordance with the present invention will now be described with reference to the accompanying drawings, in which:-

- Figure 1 is a schematic diagram of a first embodiment of recognition apparatus according to the invention;

- Figure 2 illustrates the principle of a neuron used in the apparatus, with persistence;

- Figure 3 illustrates a typical trajectory on a single array of the apparatus;

- Figures 4A and 4B illustrate the forms of soft and hard windows respectively;

- Figure 5 illustrates schematically an embodiment with stacked arrays;

- Figure 6 illustrates a localised interconnection scheme between adjacent arrays of Figure 5;

- Figure 7 illustrates an example of an array separated into two subsidiary layers to achieve intra-array grammar;

- Figure 8 illustrates a second example of an array which achieves intra-array grammar;

- Figure 9 illustrates the principle of interlayer grammar;

- Figure 10 illustrates an example specific hardware for the apparatus;

- Figures 11 and 12 illustrate a hybrid realisation of a neural element for use in the arrangement of Figure 10; and

- Figure 13 illustrates a pure analogue realisation of a neural element.

The apparatus shown in Figure 1 has a front-end speech processing unit 100. Input speech signals s are divided into frames of, for example, 10 ms duration, and each frame is analysed to generate a plurality of parameters representing characteristics of that frame. One convenient measure is the energy content of successive frequency bands within the total spectrum of the speech, which can readily be generated by transform techniques or even by a bank of band-pass filters. Such feature extraction in general and spectral analysis in particular are well known techniques for speech processing and recognition and will not, therefore, be described further here.

The output of the processing unit consists of a group of N signals (assumed here to be in analogue form), denoted by a vector $X = \{x_1, x_2, \ldots x_N\}$. Typically, N may be 16 or 32. The vectors X generated for successive frames of speech form a temporal sequence of vectors. The purpose of the remainder of the apparatus is to identify patterns (and hence phonemes, words, etc) in this sequence. Although it is described in the context of speech, it will be appreciated that recognition of patterns in other temporal sequences of vectors is also possible. The apparatus also has elements or units 101 set out in a two-dimensional array. The input to the array is supplied simultaneously to all elements of the array. Each element has a single (scalar) analogue output, denoted by η_i where is is an index number denoting the particular element. In general, η is some function of the input vector X and a weight vector $W_i = (w_{i1}, w_{i2}, \ldots w_{iN})$ associated with that element,

such as to indicate the degree of similarity between the two vectors.

Preferably, this function is the dot product, ie

$$\eta_i = W_i . X = \sum_{j=1}^{N} w_{ij} * x_j$$

Note that the dot (.) is used to signify a vector dot product and star (*) to indicate a product of scalar quantities. This function may be realised by an element of the form illustrated in Figure 2, where an amplifier AA forms an output voltage proportional to the sum of the currents in weighting resistors R_j whose values correspond to the reciprocals of stored weights w_{ij} (though a practical element will be more complex owing to the need to adjust the weights). Elements such as shown in Figure 2 are often called neurons, by analogy with biological systems, and this term is used in this description.

In order that the array can be taught the statistics of speech data, an auxiliary computer 104 is provided, which has access to the outputs η_i of the neurons and is able to change their weight vectors W_i. During training, speech is fed to the input processor and each vector X supplied to the array is dealt with in that:

(a) the computer scans the array to identify the neuron with the largest output; and

(b) it calculates, from this output and the existing weight vectors W_i, adjustments to be made to the weight vectors of the identified neuron and those adjacent to it in the array.

More specifically, such training may take the following form:

(a) initially, set the weight vectors W_i to random values;

(b) define a neighbourhood in the array around each neuron. The neighbourhood - which may be circular, square, hexagonal, etc - includes the neuron in question and a number of neighbouring neurons;

(c) take a vector X from the input. Each neuron produces an output η_i;

(d) identify - eg by scanning the neuron outputs - the neuron with the largest output (the "most excited" neuron);

(e) modify the weight vectors of all neurons lying within the neighbourhood centred on the most excited neuron using the algorithm

$$W_i' = W_i^n + k*X$$

$$W_i^{n+1} = \frac{W_i'}{|W_i'|}$$

where W_i^{n+1} is the new weight, and W_i^n is the previous weight, k is a gain factor determining the rate of adaptation; W_i' is simply an intermediate term in the calculation;

(e) repeat steps c, d and e for further vectors X.

A modification to this process avoids the necessity of explicitly identifying the most excited neuron, by applying the modification to neurons within the neighbourhood surrounding every neuron, but varying the amount of adjustment according to a non-linear function of

the output of the neuron defining the neighbourhood - eg the output raised to a high power m.

Thus, if $\eta_i' = \eta_i^m$ then the adjustment to neuron i consequent upon its lying within the neighbourhood of neuron j is:

$$W_i' = W_i^n + k*\eta_j' * X$$

$$W_i^{n+1} = \frac{W_i'}{|W_i'|}$$

In order to avoid confusion, superscripts to a symbol are here defined as denoting which of a succession of values is intended (ie W^{n-1} is the previous value of W^n): where a quantity raised to a power is intended, this will be specifically stated.

As in Kohonen's work, the rate constant k may be a function of the distance of the neuron in question from the neuron defining the neighbourhood; and it is preferred to reduce the size of the neighbourhood as training progresses.

Note that the normalisation of the weight vector magnitude to unity is an important step in this process, or else the weight vector magnitudes will tend to increase indefinitely, or (in practice) saturate. The normalisation, however, also serves another purpose. The dot product $W_i.X$ may be written as:

$$|W_i| * |X| * \cos\theta$$

where θ is the angle between the two vectors in N-dimensional space. If W_i is normalised to unity magnitude, we have

$$\eta_i = |X| * \cos \theta$$

which is the product of a term ($\cos \theta$) indicating the similarity of the input and weight vectors, irrespective of their absolute magnitude, and a term ($|X|$) which represents the input amplitude. If the outputs of an array of neurons are interpreted by considering their <u>relative</u> values, the result becomes independent of the input signal amplitude since $|X|$ is the same for all neurons.

After the array has been successfully trained, the array is ordered in terms of sound quality and the receipt of an input signal vector by the array will result in each neuron generating an output signal whose degree of excitation varies with the similarity between the input vector and the corresponding weight vector. For any given input vector, only a small group of neurons will generate output signals of any significant magnitude and these will be grouped together within the array. If the array could be viewed from above, and the neurons' output seen as lights of variable intensity, the application of a single input vector would cause a bright "blob" in one part of the array formed by the output signals from the grouped neural elements, the position of the blob being characteristic of the type of sound represented by the input vector.

Furthermore, the relationship between the array positions of the "most excited neurons" for given input vectors will (perhaps surprisingly) tend to map topologically to the relative positions in "pattern" space of those input vectors (that is, in an N-dimensional perceptual space where N is the "inherent dimensionality" of the input data). Vowels, for example, seem to be well represented by a 2-dimensional space; these types of

speech data thus have an inherent dimensionality of 2 and are well mapped onto a 2-dimensional neural array.

Note that the physical position of neurons in the array is unimportant - in the untrained array, the neurons are indistinguishable. 'Position' becomes meaningful only because of the definition of 'neighbouring' neurons.

It may be observed that a recognition apparatus may include the auxiliary computer for training, or retraining, but may lack such a computer if it can be loaded with the weights already learned by another machine.

The nature of speech sounds produced during an utterance does not change instantaneously. Consequently, vectors representing successive speech samples taken at intervals of a few milliseconds (for example 16 milliseconds) will be very similar and when applied to the array 1 will tend to cause the high output "blob" to change position by only a small amount. The effect of this is that the application of a series of input vectors representing a complete utterance will cause the blob to move along a trajectory in the array which is characteristic of the utterance. An example of such a trajectory is shown by the line of "blobs" in the array 1 of Figure 5. To obtain a continuous trajectory, it is desirable that the vectors are generated at a rate which represents an oversampling relative to the rate of change of the speech parameters which they represent.

It is important to note that the shape of the trajectory is independent of any time warp (viz variation in rate) of the utterance and so conventional pattern recognition techniques can be used to determine from the trajectory the nature of the original utterance (although the intensity profile along the trajectory is related to time warp and loudness). Figure 1 shows such a recognition unit 110.

The array described up till now produces, like that of Kohonen, only a distinct succession of high-output neurons, sometimes called "firings". Note however that use of the term firing does not <u>necessarily</u> imply a thresholding process.

The firings due to several consecutive input vectors do not co-exist and so the trajectory is not visible as a whole at one time. However, if the trajectory is to be used, it must be visible as a whole at a single instant in time. If the system is taking in continuous speech, the trajectory of firings is as long as the speech has been going on, and would eventually become infinite in length. Obviously, it would be inappropriate to retain the entire arbitrarily long trajectory and make it input to the recognition process: so the trajectory is viewed through a window in time which only passes its most recent segment as shown in Figure 3.

An appealing way of producing this window and at the same time making a complete section of the trajectory visible at a single instant in time is to build persistence into each neuron, ie once a neuron is stimulated by an input, it continues to produce an output of decaying magnitude even after the input has been removed. In general, the function may be exponential, linear, or any other function which approaches zero monotonically with time. If the decay is made exponential, then the system is using a "soft window" as shown in Figure 4 (compare with the hard window of Figure 5).

The persistence function may be defined in the following way:

$$e_i^n \;=\; e_i^{n-1} \ast \beta + \eta_i^n$$

where e_i^n is the excitation of the ith neuron at the nth sample time and β determines the effective time constant of decay. This is indicated in Figure 2 by the presence of R-C integrators 115 between the neuron outputs and the recognition unit 110. In one experiment, we have used a persistence factor β of 0.99 and a sampling period of 10 ms which yields a decay time constant of approximately 1 second.

Although the arrangement illustrated in Figure 1 is quite possible, the preferred method for achieving recognition is, as shown by the example in Figure 5, to provide a plurality of arrays; p second layers 2 are arranged in series with the first layer 1. The idea behind the stack of neural arrays is to use the trajectory produced on the bottom layer as input to the second layer, which is so connected to the first layer that it produces a more compressed trajectory therein, and so on, up through the layers of the stack until a word or even a sequence of words generates a trajectory on a high layer which is so short that it can be associated with just one or a small group of neurons. The final recognition stage 110' then serves simply to identify that one or small localised group of neurons having the largest output. This neuron can be labelled explicitly with the appropriate utterance.

It is then necessary to devise a scheme to present the windowed section of the trajectory on one array to the neurons of the next array.

One obvious solution would be to make a very large vector from the outputs of every neuron in one layer and present it as input to every neuron in the next layer.

However, for inter layer connections, this arrangement
is unsuitable on two counts. First, there may be several
tens of thousands of neurons in layer n, giving rise to an
input vector to layer n+1 neurons whose dimensionality
would also be tens of thousands - this would require
around a billion inter layer connections, which is simply
not physically possible.

The second reason for avoiding this scheme is that the
inherent dimensionality of firing patterns derived from
the whole of a layer is almost certain to be much greater
than two, and hence it would be impossible to map them in
an ordered manner onto the next layer which only has two
dimensions.

A solution avoiding both of these problems is
illustrated in Figure 6. Here the input vector to a
neuron in a layer 3 and having co-ordinates (x,y) is
derived from the outputs of a fairly small number of
neurons in a layer 4 which lie within a small circle whose
centre has the same coordinates (x,y). The circular
regions in the layer 4 overlap, so that the inputs to
neurons within a small locality in the layer 3 will be
very similar, and the array in the layer 3 will become
ordered after training in a similar way to an array which
has a global input vector, except that the ordering will
be local rather than global.

This connection strategy overcomes the problem of the
numbers of interconnections. If layer 3 contains 10,000
neurons each of which is connected to 100 neuron outputs
in the previous layer, then 10^6 interconnections would
be needed.

The developed neural array preferably has decaying
persistence built into each element and this causes the
trajectory of excited elements to be partially "frozen" on

the array, such that at one instant in time the most recently stimulated element on the trajectory is most excited and all the other elements along the trajectory are also excited but with decreasing intensity. Thus the system described is able to transform a sequence of pattern vectors representing a process with variable time warp, to a single pattern of element excitation in the neural array. This single pattern is time warp independent if the excitation intensity profile along the array is ignored.

The patterns of excitation in the array can be expressed as a vector in many other ways, but, for example, could be a vector whose elements were the co-ordinates of the z most intensely excited elements in the array where z is some fixed number. Thus, another way in which the inputs to the neural elements of one layer may be derived from the outputs of the previous layer is as follows:

The outputs of all the neurons in the lower layer are thresholded with a threshold which has a value such that only those elements which lie approximately on the path of trajectories of excited elements in the lower layer produce an output. Alternatively, another form of non-linearity is made to act on the outputs of all elements in the lower layer. For example, the output of each element may be raised to the power m where m is greater than one. Imposing this type of non-linearity on the element outputs has the effect of accentuating the difference in excitation between elements which lie in the vicinity of a trajectory and those that do not lie near a trajectory.

Consider now the required persistence of a recogniser with two arrays.

If the time constant was made long, say several seconds, then the first layer trajectory would cover several words. The numbers of allowable permutations of even a few words taken out of a moderately sized vocabulary is immense and so there would be an enormous number of different trajectories, each of which would need to cause strong firing of a distinct neuron in the subsequent layer. The number of neurons required in the next layer would be unrealisable, either artificially or biologically.

At the other extreme, if the time constant is made so short that only the firing of the last neuron in the first layer was retained, no compression of trajectory in the second layer would be possible because the first layer has merely recoded the sequence of speech input vectors as a sequence of fired neuron co-ordinates - a 1:1 re-coding.

A sensible choice of time constant seems to be a little less than the length of the shortest word that will be encountered, about 200 mS. The consequence of this choice is that the trajectory produced by the shortest word will cause a trajectory on the second layer which has near zero length, ie just one small group of neurons will fire. Clearly, an alternative method would be to build the persistence into the inputs of the neurons in the second layer.

Where there are three layers, it may be convenient to build into the outputs of the second layer a longer persistence than exists in those of the first.

In further embodiments, certain concepts of grammar are built into the system. One method which we have devised for achieving this recognises that some trajectories are more likely to occur than others, and therefore biases the trajectories towards those preferred

trajectories by cross-coupling within the layer. We separate at least the first layer 1 of Figure 5 into two parts. The first part 5 (Figure 7) is just a standard array with persistence so that at the nth sample instant the output of the ith neuron in this system is e_i^n :

$$e_i^n = \eta_i^n + e_i^{n-1} * \beta \quad \ldots$$

The outputs of neurons in this array are fed after weighting (in multipliers 7) by stored factors of a_{ij} into the second part 6, an array of adding nodes 9 which sums e_i for each neuron with the weighted outputs $e_{i,j'}$ of the q neurons within a neighbourhood of the neuron in the previous array to give an overall neuron

excitation of e_i' :

$$e_i'^n = e_i^n + \sum_{j=0}^{q-1} e_{i,j}^{n-1} * a_{i,j}$$

where $e_{i,j}$ is the excitation of the jth one of the q neighbouring neurons.

The factors a_{ij} comprise grammar weights which tend to guide utterance trajectories along likely paths in the array, these paths being those expected in conventional speech. This therefore makes the shape of the path less dependent on speaker variation or background noise. Conceptually, the first part may be considered to be a topologically ordered array of "receptor neurons" while the second part 6 may be considered to be an array of "ganglion" neurons. It should be noted that the second part 6 is not ordered in the normal sense. Note that in Figure 7, e_{ij}^n is used rather than e_{ij}^{n-1} . Whether this is possible in practice will depend on the delays in the system.

- 19 -

The advantage of this arrangement is that it assists in enabling the system to learn primitive grammar.

The system works as follows: assume for the moment that all the weights, a, between ganglions and receptors are zero. Say that a common sequence of excitation in the receptor array 5 is neuron R1 followed by neuron R2 (R denotes receptor and G ganglion). Excitation of R1 followed by R2 in the receptor array causes excitation of nodes G1 and G2 in the ganglion array 6. If the second input pattern to the receptor array 5 is corrupted in some way such that neuron R4 is fired instead of R2, then node G2 in the ganglion array will be correspondingly excited. Thus the system is unable to combat corruption or noise in the input patterns.

Assume now that $a_{1,2}$ has a large value.

Consider what happens when the second corrupted input pattern is now applied to the receptor array 5: it excites neuron R4 which tends to excite node G4 in the ganglion array 6. However, the previously excited receptor neuron R1 is still excited due to persistence and is sending a strong signal via $a_{1,2}$ to neuron G2 causing it to be excited. Thus the overall sequence of excitation in the ganglion layer will tend towards G1 followed by G2 even though the input was corrupted.

The receptor to ganglion connection weights are learnt from the application of a large number of pattern sequences. A typical learning algorithm might be:

$$a_{i,j}^{n} = a_{i,j}^{n-1} * \gamma + k_2 * e_i^{n} * e_{i,j}^{n-1}$$

where:

      n is the sample time index,

      γ is a forgetting factor,

      k_2 is a learning factor,

      e_i and $e_{i,j}$ are as defined above.

This can be implemented by the same auxiliary computer which deals with the weight vectors W_i, and in a similar manner.

An alternative embodiment with means for introducing grammar into the system is illustrated schematically in Figure 8. In this case, neurons i, j of each array are coupled by weights $a_{i,j}$, $a_{j,i}$ with their immediately surrounding neighbours. By suitably setting the values a, a trajectory within a particular layer will follow a preferential route since the moderate stimulation of a neuron will cause large stimulating signals to be fed via the neural interconnections having large weights to the neighbouring neurons which will respond to these signals when processing the next input vector.

A typical learning algorithm for this system might be:

$$a_{ij}^{n+1} = a_{ij}^{n} * \gamma + k_2 * W_i \cdot X^{k-1} * W_j \cdot X^k$$

where γ, k_2, are as defined above, n is the update rate of the weight factors a and k is the update rate of the input vectors X.

In many cases, the trajectory produced by an utterance in the first layer 1 will be substantially continuous. However, not all trajectories will be fully continuous and will instead consist of a series of separated relatively short segments. This might occur, for example, in the first layer, as it receives a vowel-stop consonant-vowel $(V_1 - C - V_2)$ utterance.

There will be no neural firing during the stop consonant because it is by definition silence. The two vowels on either side of the stop will not in general be phonetically adjacent, and so, the trajectories they generate will not be contiguous on the array. Nevertheless, the allowable V_1-C-V_2 sequences in a particular language are limited and it is desirable to exploit this constraint such that the occurence of a trajectory associated with (V_1-C) is used to "prime" the neurons which would lie on the trajectory for (C-V_2) and make them more likely to be excited than other groups of neurons.

A scheme is shown in Figure 9 in which the global firing pattern in a higher layer 9 is fed back and presented as an input vector to the layer 8 below along with the ordinary input vector. The scheme would operate on the (V_1-C) and (C-V_2) trajectories in the following way. The occurrence of the V_1-C trajectory on the layer 8 causes a compressed trajectory to occur on the layer 9 due to the forward inter-layer connection. This compressed trajectory is encoded in a vector, F which is fed back, with delay, to the input of the layer 8 where it is appended to the ordinary input vector, X, to form an overall input to the layer 9 of (X:F) = Q. In Figure 9, the delay 10 is shown as a separate element in the system but in reality it is expected to be implicit in the neural arrays themselves. If the feedback delay is similar to the time of the stop between V_1 and V_2, then F actually gets to the input of the layer 8 at the same time as the vector X representing the start of V_2. Assuming that the self organisation of the layer 8 takes place with the feedback path active, its presence should become a necessary input component to cause the trajectory of

$(C-V_2)$ to be generated, i.e. the input vector X with
null feedback vector would only cause weak excitation of
the $(C-V_2)$ trajectory, as would a null input vector
appended by an active feedback vector. However, the <u>joint</u>
occurrence of an active feedback, F, and input X should
cause high excitation of the $C-V_2$ trajectory, since this
is the input on which the system has been trained.

This feedback also has the result that recognition of
a particular word will predispose lower layers to
recognise the primitive sounds associated with the
anticipated next word. The purpose of this
predispositions to aid recognition of ill defined input
patterns.

In an ideal system, the output from each layer would
be fed back as an input to every previous layer. However,
in practice this would lead to a totally unmanageable
number of connections and inputs to each neuron. To
achieve an approximation of this ideal arrangement, in the
preferred example all or a part of each output signal from
each layer above the first is fed back to the preceding
layer so that the output of a high layer indirectly
affects the inputs to all previous layers.

The operation of the system can be understood by
considering a higher layer in which words are recognised
and a lower layer in which phonemes are recognised. Say
that a particular word has just been recognised in the
word layer. This information is fed back and forms part
of the input to the phoneme layer at the same time as the
patterns corresponding to the phoneme of the next word are
coming in.

The phoneme layer is trained on the combination of
feedback pattern plus input pattern and so both are
necessary to cause full excitation of the phoneme array
neurons associated with the phoneme in the second word.

Now, imagine that the second word is heavily corrupted
or does not even arrive as input to the phoneme layer.
This layer will still be getting a partial input from the
pattern fed back from the word layer and so will tend to
excite neurons which would have been fully excited by the
succession of phonemes in the second word if it had
arrived.   Thus,   weak   excitation   representing   the
non-existent or corrupted second word will be generated.
This form of error correction using knowledge of what is
likely to happen in a sequence of patterns aids
recognition of ill defined patterns.

These methods cannot be implemented in real-time for
processes such as speech recognition except by using
parallel processing arrays.

Figure 10 illustrates the architecture of a neural
array structure having m neural or computing elements N_1
... N_m, each of which is associated with a respective
local RAM 20.   Each RAM stores synaptic weights for the
associated neural element.

It should be noted that the arrangement shown in
Figure 10 enables the output signals from each neural
element of one array to be supplied to each neural element
of the next array.

In Figure 10 one array 21 is shown in full while parts
of the preceding array 22 and the succeeding array 23 are
also shown.

Before explaining how the system works, define the
required output of layer 21.

$$\eta_i^n = \eta_i^{n-1} * \beta + \sum_{r=1}^{m} x_r * w_{ir} + \sum_{r=1}^{m} y_r * c_{ir}$$

where x is the input vector from the layer 24 below (or,
for the first layer, the input vector), y_r is the

feedback vector from the layer 23 above, and β, w_{ir} and c_{ir} are the persistence weight and feedforward and feedback synaptic weights respectively contained in the local RAM.

Operation of the system is as follows: Assume that the neurons N_1 to N_m have just finished computing their respective outputs η_1^{n-1} to η_m^{n-1} which are stored in respective accumulators of the neural elements (not shown) and switch S_2 is set at position b. Switches S_{31} to S_{3m} are set to position a so that the neural outputs are loaded into the storage elements (T) of the output bus 24 of layer 21. At the same instant switches S_{11} to S_{1m} are set to position b so that each neuron's output η_i^{n-1} is connected to its input along the persistence path 25 and the accumulators are cleared. At this instant all the neurons start computing the first component of their output – ie the persistence terms, which for neuron N_i is $\eta_i^{n-1} * \beta$. So the neurons have just worked out the first term in the above expression which is stored in respective accumulators. β is obtained from the RAM 20 associated with each neuron.

Switches S_{31} to S_{3m} of all the arrays now move to position b and switches S_{11} to S_{1m} switch to position a such that the first value, x_1, stored on the feed forward bus of layer 22 is applied to the inputs of all neurons in layer 21 simultaneously while the appropriate synaptic weights w_{11} to w_{m1} are drawn from each neuron's local RAM. (At the same time, the first value on the feed forward bus of layer 21 is being applied to the inputs of neurons in layer 23 and so on, up through the stack of arrays).

At this point the next component in each neuron's output is computed within the neuron and is $x_1 \cdot w_{i1}$ and is added to the contents of the accumulator.

The feed forward bus 24 on each array is then "clocked" along one place so that the next value, x_2, from the array is applied to all neurons in the next array, and the next partial output $x_2 \cdot w_{i2}$ is computed and added to the previous partial output in the accumulator of each neuron. This process is continued until the feed forward bus has been completely circulated.

Next switch S_2 is moved to position _a_ and the process is repeated with y_1 to y_m is turn as inputs. These are values fedback from the output bus of a higher layer along feedback path 26.

On completion of this process each neuron has computed a new output value η_i^n stored in the accumulator and the entire process re-commences.

The non-linearity processor 27 in series with the output bus 24 of each array is optional. Its function, if included, is to threshold, or raise to a power, the values stored on the bus.

The system shown in Figure 10 could be realised digitally with all synaptic weights being represented, for example, by eight bit words and the neuron output and input signals also being represented by eight bit words. The circulation of data on the output bus, feedback path and persistence paths would all be done serially to minimise the amount of hardware associated with each neuron and also the number of interconnection conductors between neurons. Therefore each storage element, T, in the output bus 24 will consist of an eight bit SISO shift register.

Figure 10 tacitly assumes that the output of all the
neurons of an array form the input for the next layer. As
explained above, this is often unrealistic; however, the
architecture of Figure 10 can readily be modified by
connecting groups of neurons to separate buses
corresponding to the local groupings illustrated in
Figure 4.

Figure 11 illustrates a hybrid analogue/digital
realisation of a computing element which may be used in
the arrangement of Figure 10. In this case, all synaptic
weights are stored digitally in the RAM 20 which is
addressed from a control circuit 30 via an address bus
31. Digital values from the RAM 20 are fed to a
multiplying digital-to-analogue converter (DAC) 32 which
also receives via an input line 33 the analogue input
signal from the respective switch S_{11}-S_{1m}. Operation
of the DAC 32 is controlled by clock signals generated by
the control circuit 30 and fed along a control line 34.
The DAC 32 may be realised for example by a switched R-2R
ladder or ramp comparator. In this case, the accumulator
35 associated with the computing element is formed by a
R-C leaky integrator.

In this example, each storage element T in the serial
bus will be realised by a single stage clocked analogue
shift register to which clock signals are supplied from
the control circuit 30. This shift register could be a
charge-coupled device, a bucket brigade line or sample and
hold circuit. In addition, all signal values in this
system will be represented by analogue voltages and be
circulated on the serial output bus 24 and feedback paths
in pulse amplitude modulation.

Figure 12 illustrates a modification of the Figure 11
example in which the accumulator 35 is formed by a
resettable analogue integrator with a switch 36 controlled
via a control line 37 by the control circuit 30.

Figure 13 illustrates a pure analogue realisation of a neural element. In this version, the serial bus structure of Figure 10 is abandoned. It is assumed that the outputs of neurons in one layer are physically connected to the inputs of neurons in the next layer. The voltages representing each element of a pattern vector applied to the bottom array in the stack are applied in parallel to the inputs of each neuron in that array, i.e. if the pattern vector has 30 elements, then each neuron is the bottom array must have 30 parallel inputs. The analogue circuitry enables the synaptic weight to be updated in real time without the use of an auxiliary computer. The system is entirely self-sufficient.

The signal inputs to the neuron are $x_1 \ldots \ldots x_N$.

These voltages cause currents I_1 to I_N to flow into a summing point P. The magnitude of each current is

$$I_j = x_j / R_{DSj}$$

where R_{DSj} are the drain source resistance of transistors T_{2j}. The factor $1/R_{DSj}$, is therefore effectively the jth synaptic weight of the neuron and the total current at the summing point P is therefore proportional to the output of the neuron. (This total current is converted to a voltage by passing it through a resistor R_3).

The magnitude of the resistance R_{DSj} is controlled by the voltage applied to the gate of transistor T_{2j} which in turn is controlled by the current flowing through T_{1j}. The current through T_{1j} depends on the amount of charge stored on its gate which in turn is supplied via T_{3j} from the signal input voltage x_j.

Thus, if x_j is frequently large, the gate of T_{1j} will become charged, causing increased current flow, i_j, which increases the voltage on the gate of T_{2j}, reducing the drain-source resistance of this transistor and thereby increasing the effective value of the jth synaptic weight. For the whole neuron, the synaptic weight vector is modified after a time $\Delta t+$ to become:

$$W(t + \Delta t) = W(t) + L\Delta t.$$

Now, according to the ordering algorithm for the array, only synaptic weight vectors of those neurons which are in the neighbourhood of the most strongly excited neurons should be modified. This is achieved in this circuit by controlling the flow of current to the gate of T_{1j} by using the sum (from a summation unit A) of the output of the neuron and the outputs of adjacent neurons in the array (ie those in whose neighbourhoods the neuron lies) to control the voltage on the gate T_{3j} which controls the flow charge onto the gate of T_{1j}. In other words, the factor L in the previous expression is the aggregate excitation of all the neurons in a particular part of the array.

A second point to consider is that the ordering algorithm specified only works if the magnitude of the synaptic weight vector is always kept at unity. This is achieved in the analogue circuit by deriving the currents I_1 to I_N from a single current source, I, such that:

$$I = \sum_{j=1}^{N} I_j$$

The synaptic weights are proportional to I_j and so

$$\sum_{j=1}^{N} w_j = \text{constant}$$

This means that the magnitude of the weight vectors is a constant in a City Block sense rather than Euclidean sense. However, simulation has shown this to be satisfactory.

Several learning processes have been discussed above. Although in principle it may be possible to run at least some of them simultaneously, it is envisaged that a more orderly approach would be to follow a sequence such that the neuron weights W are learned before the grammar weights a (if used), and that each layer is trained before a higher layer is trained. Thus we may have a sequence:

    (a) learn weights W_i of first layer.

    (b) learn weights a_{ij} of first layer.

    (c) learn weights W_i of second layer.

    (d) learn weights a_{ij} of second layer.

    (e) learn feedback weights c between second and first layers (if feedback is provided).

    (f) continue for third and subsequent layers.

CLAIMS

1.   A pattern recognition apparatus comprising:
     - an input for receiving a temporal sequence of input
signal vectors;
     - store means for storing a plurality of reference
vectors;
     - a like plurality of comparison elements arranged in
operation to receive the same, current, input vector and
to generate an output signal indicative of the degree of
similarity between the input vector and a respective one
of the reference vectors;
     - means for producing for each element an output
modified in dependence on the output signal produced by
that element as a result of a comparison between its
reference vector and at least the immediately preceding
input vector; and
     - recognition means for comparing the pattern
represented by those of the said modified outputs which
indicate a relatively high similarity between the input
and reference vectors with reference information to
identify patterns represented by the said temporal
sequence.

2.   A pattern recognition apparatus according to
claim 1, in which the modification means is arranged to
produce for each element an output which is the sum of the
current output of that element and a plurality of earlier
outputs of that element, each multiplied by a value which
is a monotonically decreasing function of the time which
has elapsed between that earlier output and the current
output.

3.   A pattern recognition apparatus according to claim 2, in which the function is a decaying exponential and the modified signal is given by:

$$e_i^n = e_i^{n-1} * \beta + \eta_i^n$$

where e_i^n and η_i^n are respectively the modified and unmodified outputs of the ith element at the nth sample time and β determines the effective time constant of decay.

4.   A pattern recognition apparatus according to claims 1, 2 or 3, in which the store means contain reference vectors generated by:
    - applying to the input a training sequence of signal vectors;
        - for each input vector:
            -    identifying that element which has the output indicating the greatest similarity between input and reference vectors;
            -    applying to the reference vectors of the identified element and those elements lying, in a notional array space, within a defined neighbourhood of that element an adjustment in a sense such as to increase the similarity between that weight and the input signal vector,
whereby the reference vectors represent a topological mapping of the statistics of the training sequence.

5.   A pattern recognition apparatus according to
claims 1, 2 or 3, including control means arranged in
operation during the application of a training sequence of
signal vectors to the said input to perform the steps of:
-    for each input vector:
-    identifying that element which has the
output indicating the greatest similarity
between input and reference vectors;
-    applying to the reference vectors of the
identified element and those elements lying,
in a notional array space, within a defined
neighbourhood of that element an adjustment
in a sense such as to increase the
similarity between that weight and the input
signal vector,
whereby the reference vectors represent a topological
mapping of the statistics of the training sequence.

6.   A pattern recognition apparatus according to any
one of the preceding claims, including means for producing
for each element a further modified output which is the
sum of the modified output of that element and a weighted
sum of the modified outputs of other elements.

7.   A pattern recognition apparatus according to
claim 6, in which the weights employed in forming the said
weighted sum have been generated by applying a training
sequence of signal vectors to the input and iteratively
adjusting the value of the weight to be employed between
each element and each of said respective other elements by
an amount dependent on the product of outputs produced by
those two elements.

8.   A pattern recognition apparatus according to
claim 6, including control means arranged in operation
during the application of a training sequence of signal
vectors to the said input to iteratively adjust the value
of the weight to be employed between each element and each
of said respective other elements by an amount dependent
on the product of outputs produced by those two elements.

9.   A pattern recognition apparatus according to any
one of the preceding claims, in which the said modified or
further modified output are subjected to a non-linear
operation such as to enhance values indicative of high
similarity relative to these indicative of lower
similarity.

10.   A pattern recognition apparatus according to any
one of the preceding claims, in which the recognition
means comprises at least one further array of such
comparison elements having respective stored reference
vectors, each of which elements is connected to receive a
respective group of the modified (or further modified)
outputs of the or a preceding array so as to generate an
output signal indicative of the similarity between its
reference vector and the said group of outputs.

11.   A pattern recognition apparatus according to
claim 10, arranged to feed back the output signals
generated by one or more of the arrays so as to modify the
input signal vector, or the groups of signals, fed to the
preceding array.

12.   A pattern recognition apparatus according to any
one of the preceding claims, in which the reference

vectors are normalised such that the identity of that
element which produces an output indicative of the
greatest similarity between the input vector and its
reference vector does not vary with the magnitude of the
input vector.

13.   A pattern recognition apparatus according to
claim 12, in which each element is such as to produce an
output which is, or is a function of, the dot product of
the respective reference vector and the input vector.

14.   A pattern recognition apparatus comprising:
    - an input for receiving a temporal sequence of input
signal vectors;
    - an array comprising a plurality of comparison
elements each having an output and a plurality of inputs
connected in parallel with the inputs of the other
comparison elements so that each receives the
instantaneous input vector;
    - store means for storing a like plurality of
reference vectors; each comparison element being arranged
in operation to produce at its output a signal indicative
of the similarity between the input vector and a
respective one of the reference vectors;
wherein the output is, or is a function of, the dot
product of the input and reference vectors and the
reference vectors are normalised such that the identity of
that element which produces an output indicative of the
greatest similarity between the input vector and its
reference vector does not vary with the magnitude of the
input vector.

15.   A pattern recognition apparatus according to claim 13 or 14, in which each element, for receiving an N-dimensional signal vector, comprises a plurality N of sections each having:

- an input
- a variably conductive path between that input and a current summing node to form an output;
- a storage transistor consisting of an insulated gate field effect transistor arranged to control the conductivity of the path in dependence on the amount of charge stored on the gate of the second transistor;
- means for adjusting the said stored charge in dependence on the output of the element and the outputs of neighbouring elements.

16.   A pattern recognition apparatus according to claim 15, in which the conductive paths are formed by a further transistor and the storage transistors are connected to supply a proportion of the current from a single constant current source to arrangements for controlling the conductivity of the further transistors, whereby the algebraic sum of their conductivities is substantially constant.

17.   A speech recogniser according to any one of the preceding claims, including speech processing means operable to generate from each of successive periods of speech input thereto a plurality of parameters forming the components of the said signal vector.

18.   A pattern recognition apparatus according to claim 17, in which the duration of the periods of speech is such as to represent an oversampling of the speech relative to the rate of change of the said parameters.

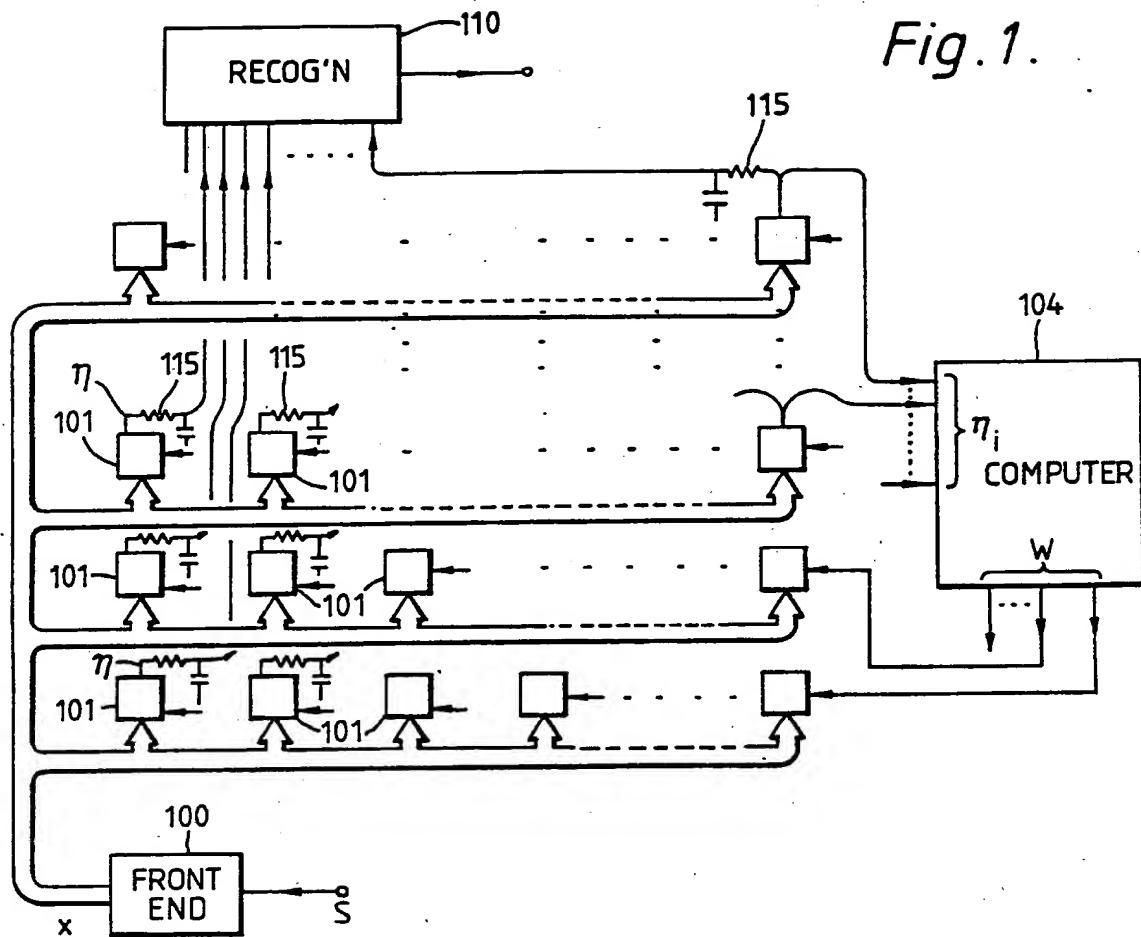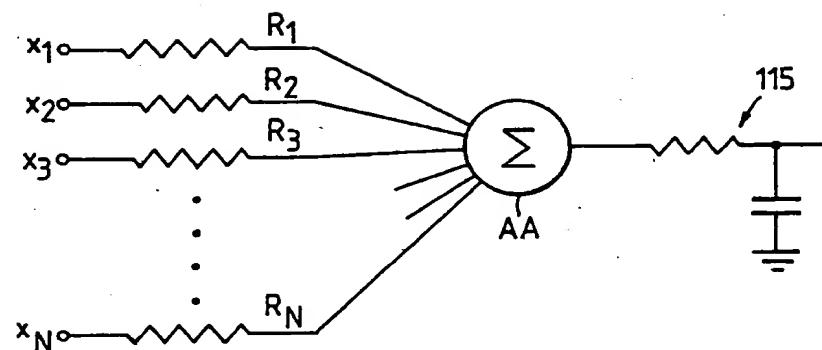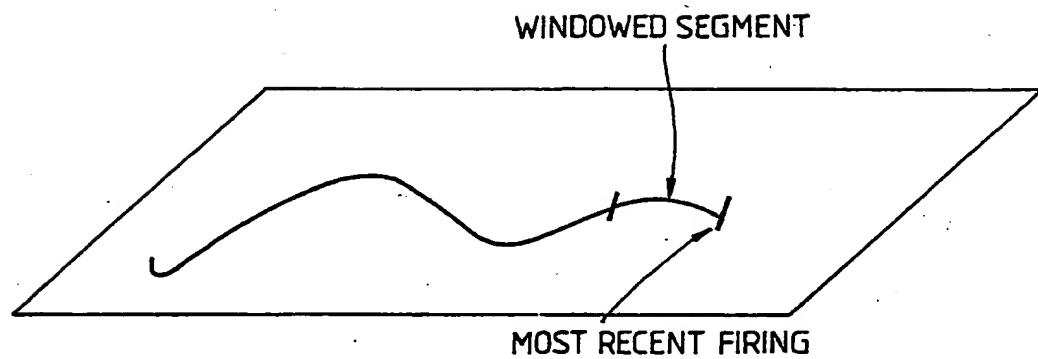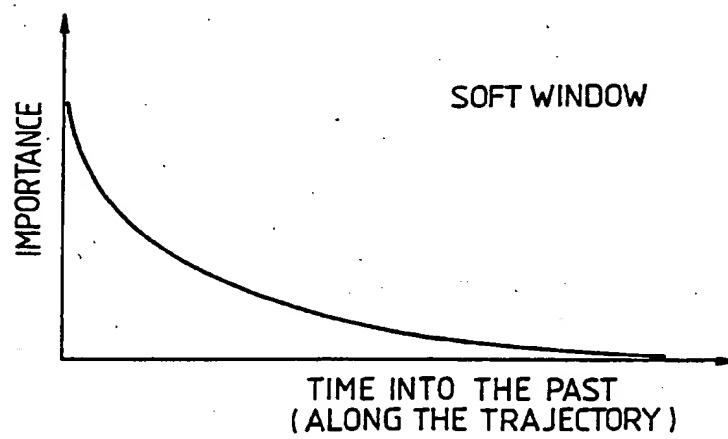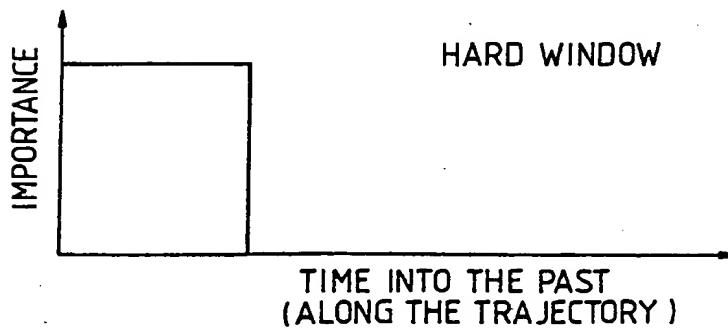*Fig.1.*

Fig.2.

Fig.3.

WINDOWED SEGMENT

MOST RECENT FIRING

Fig.4a.

SOFT WINDOW

IMPORTANCE

TIME INTO THE PAST
(ALONG THE TRAJECTORY)

Fig.4b.

HARD WINDOW

IMPORTANCE

TIME INTO THE PAST
(ALONG THE TRAJECTORY)

Fig.5.

110'

2ND LAYER p

2

2ND LAYER 1

2

1ST LAYER

INTER-LAYER CONNECTION

1

N-D INPUT VECTORS

SUBSTITUTE SHEET

Fig.6.

Fig.7.

Fig. 8.

Fig. 9.

LAYER 9

DELAY

10

F

LAYER 8

INPUT VECTORS

Fig.10.

7/8

Fig.11.

Fig.12.

$\frac{8}{8}$

Fig. 13.

FROM OTHER NEURONS

S
NEURON OUTPUT

FIRST ELEMENT
OF INPUT PATTERN
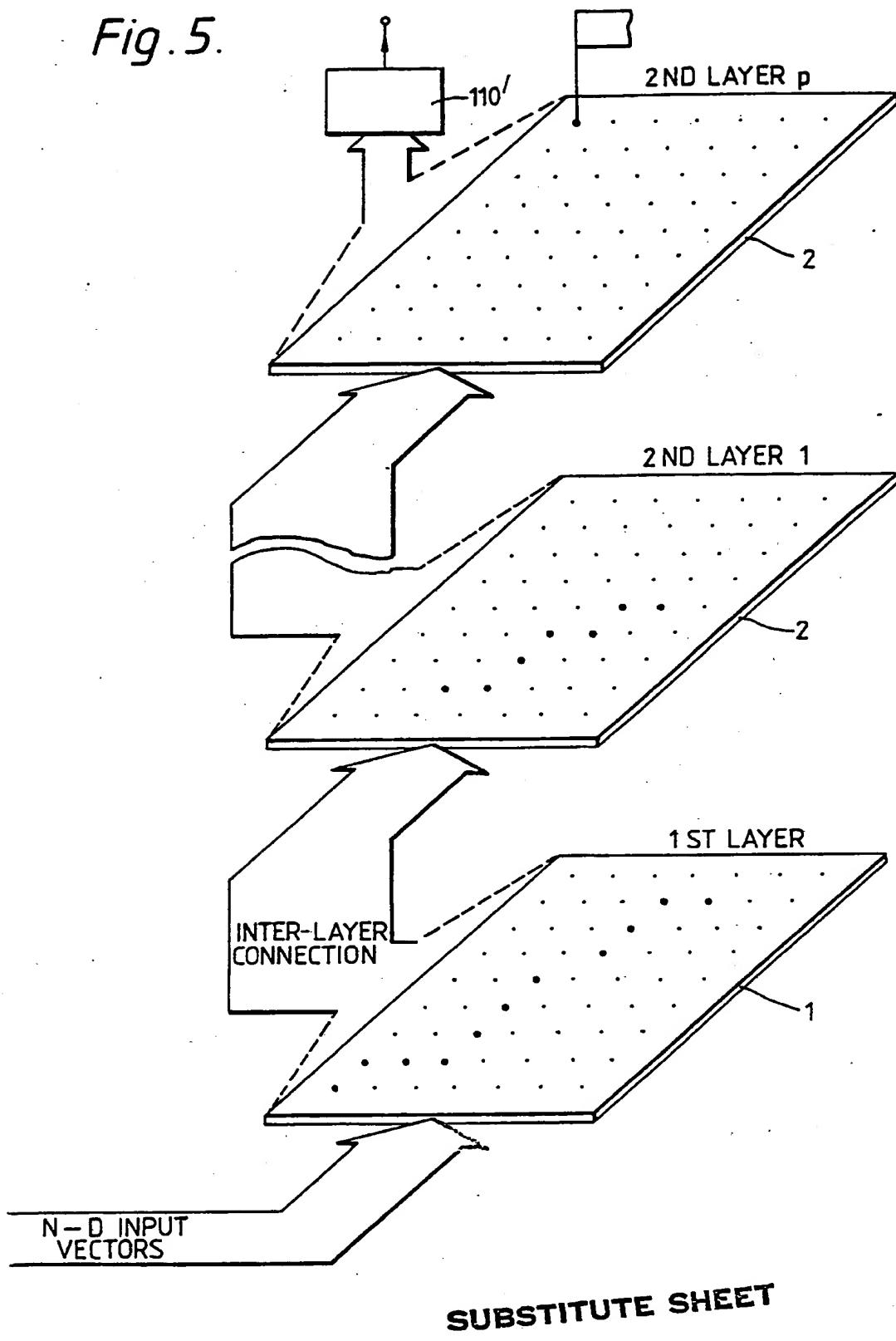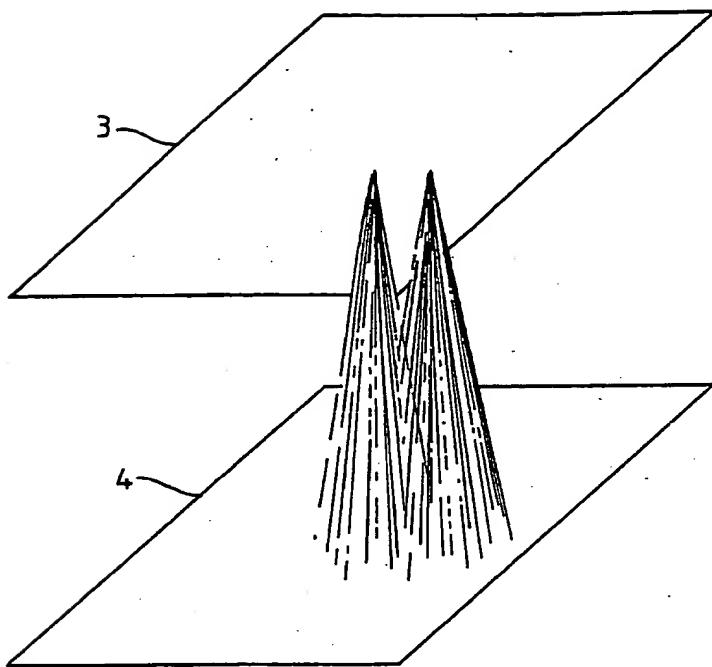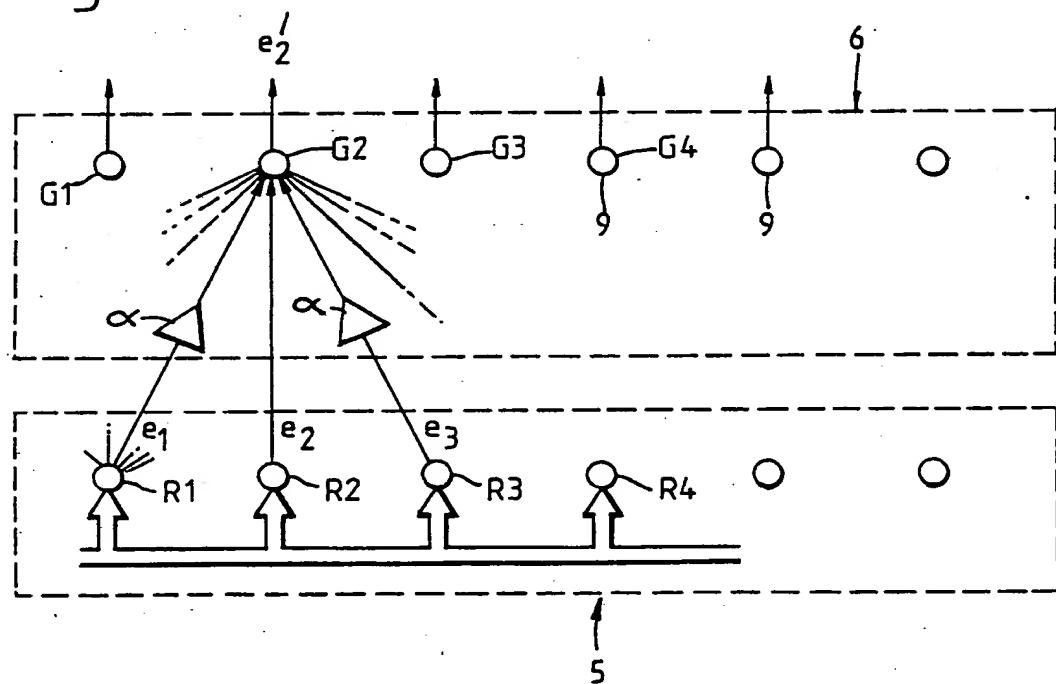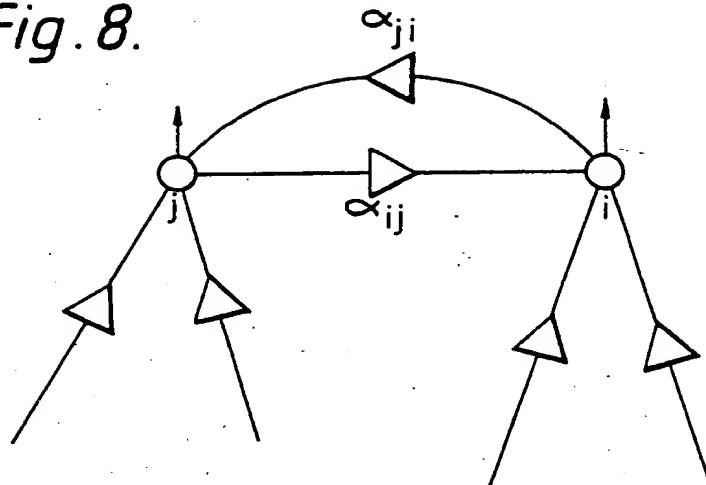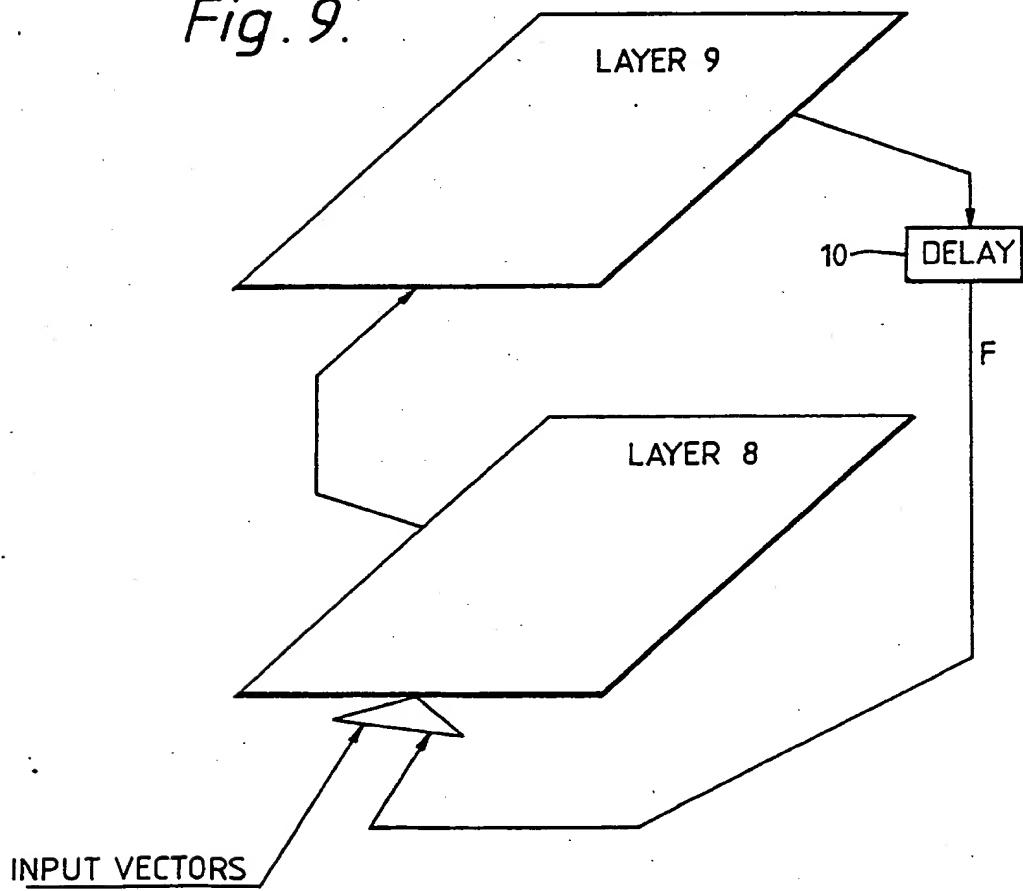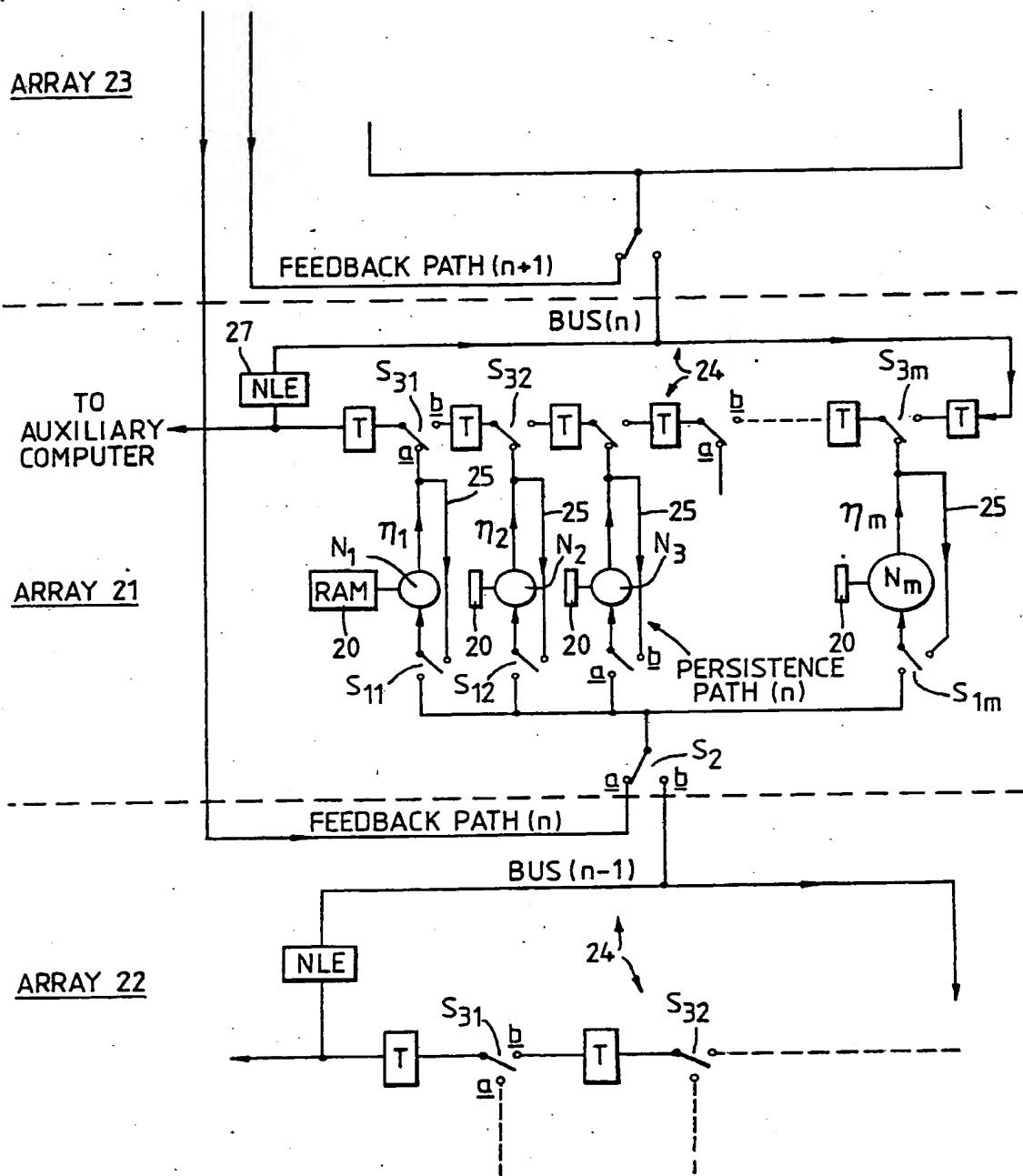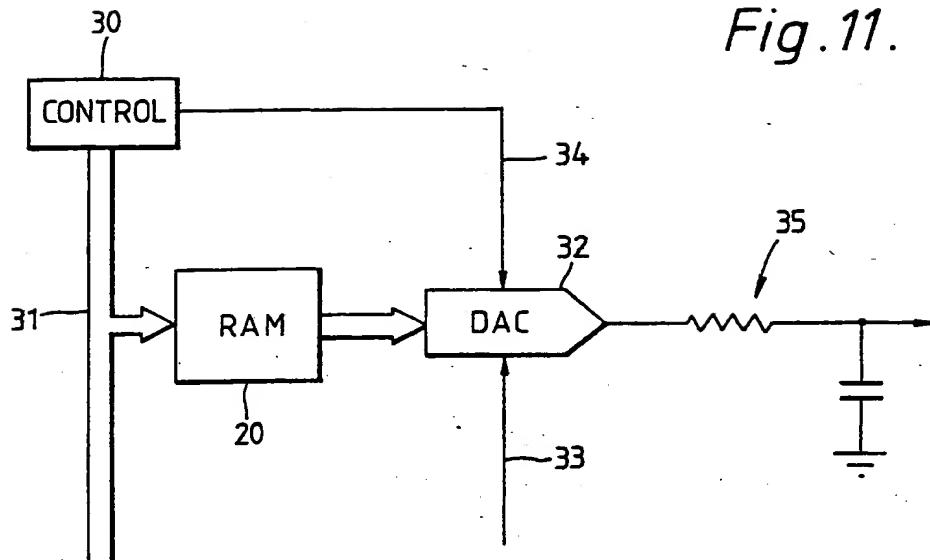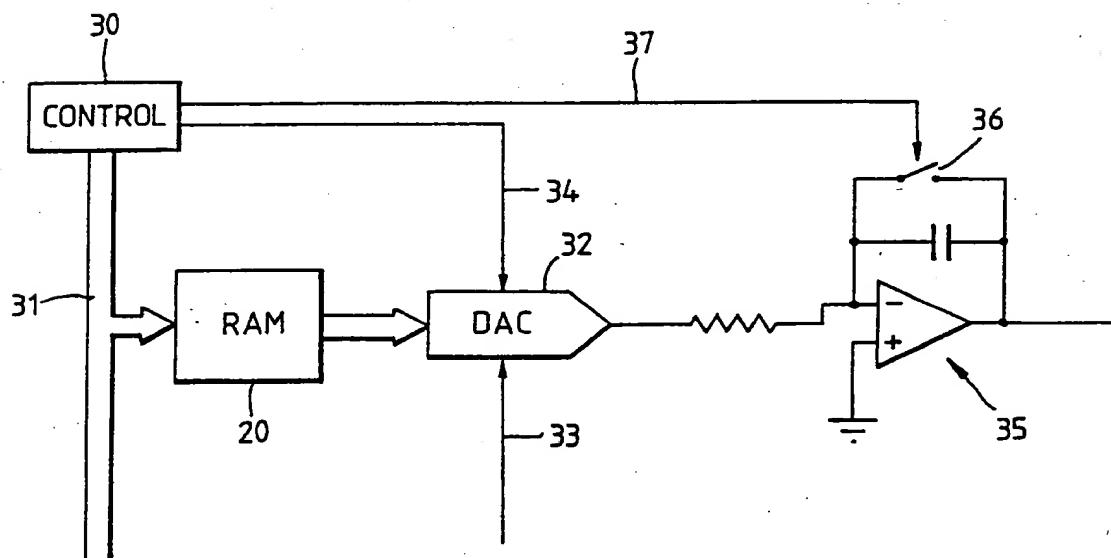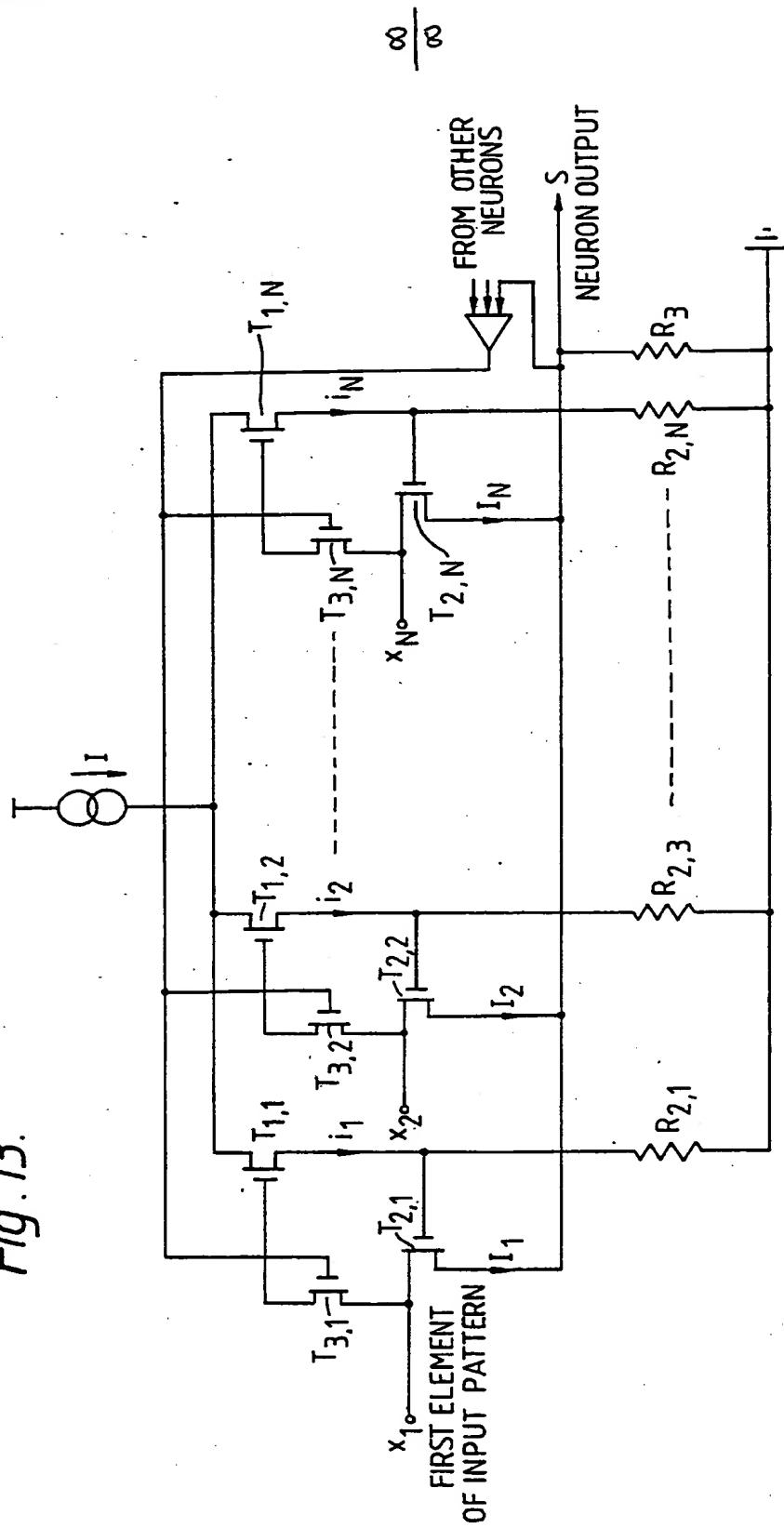
INTERNATIONAL SEARCH REPORT

International Application No   PCT/GB 88/00710

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) *

According to International Patent Classification (IPC) or to both National Classification and IPC

IPC⁴: G 06 K 9/66

II. FIELDS SEARCHED

Minimum Documentation Searched 7	
Classification System	Classification Symbols
IPC⁴	G 06 K; G 06 F; G 06 G; G 10 L

Documentation Searched other than Minimum Documentation
to the Extent that such Documents are Included in the Fields Searched *

III. DOCUMENTS CONSIDERED TO BE RELEVANT *

Category *	Citation of Document, 11 with indication, where appropriate, of the relevant passages 12	Relevant to Claim No. 13
Y	IEEE Proceedings of the 6th International Conference on Pattern Recognition, 19-22 October 1982, Munich, IEEE, (New York, US), T. Kohonen: "Clustering, taxonomy, and topological maps of patterns", pages 114-128 see page 114, paragraph: ""On the representation of complex empirical data"; page 116, right-hand column, line 22 - page 119, left-hand column, line 35, page 123, paragraph: "Formation of phoneme maps"; figure 4 cited in the application	1-18
Y	EP, A, 0085545 (K.K. TOSHIBA) 10 August 1983 see abstract; page 6, line 16 - page 7, line 31; figures 1,3a	1-18
Y	Transactions of the I.R.E., Professional Group on Information Theory, no. 4, September 1954, ./.	1-3,9

* Special categories of cited documents: 10

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
22nd November 1988	2 9. 12. 88

International Searching Authority	Signature of Authorized Officer
EUROPEAN PATENT OFFICE	T.K. WILLIS

Form PCT/ISA/210 (second sheet) (January 1985)

III. DOCUMENTS CONSIDERED TO BE RELEVANT   (CONTINUED FROM THE SECOND SHEET)

Category*	Citation of Document, with indication, where appropriate, of the relevant passages	Relevant to Claim No
	B.G. Farley et al.: "Simulation of self-organizing systems by digital computer", pages 76-84 see page 80, lines 1-20 --	
A	L'Electricité Electronique Moderne, volume 42, no. 265, June/July 1972, B.H. Marin: "Le neurone cybernétique", pages 21-25 see page 22, left-hand column, lines 16-33; figure 17a; figure 20: "Contrôle-rétroaction" --	1
A	Electronics Letters, volume 4, no. 20, 4 October 1968, I. Aleksander et al.: "Microcircuit learning nets: improved recognition by means of pattern feedback", pages 425-426 see figure 1 --	1
Y	Pattern Recognition, volume 15, no. 6, 1982, Pattern Recognition Society, K. Fukushima et al.: "Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position", pages 455-469 see page 457, right-hand column, lines 8-16; page 464, paragraph 5; figures 2-4,8	1,6,10
A	--	7,8
Y	WO, A, 87/02805 (NHK) 7 May 1987 see abstract; page 4, line 8 - page 5, line 8 & EP, A, 0245508 --	1,10,11
Y	IEEE Transactions on Computers, volume C-20, no. 9, September 1971, IEEE, (New York, US), L.C.W. Pols: "Real-time recognition of spoken words", pages 972-978 see page 975, right-hand column, lines 28-32 --	1,12,14
Y	GB, A, 1192554 (ELECTRIC & MUSICAL INDUSTRIES) 20 May 1970 see page 2, lines 31-53; page 3, line 53 - page 4, line 27; figures 1-3 --	1,10

./..

Category*	Citation of Document, with indication, where appropriate, of the relevant passages	Relevant to Claim No
Y	IEEE Journal of Solid-State Circuits, volume SC-1, no. 2, December 1966, IEEE, (New York, US), J.W. McConnell et al.: "MOS adaptive memory elements as weights in an adaptive pattern classifier", pages 94-99 see page 95, paragraph: "Adaptive element"	15
X,P	British Telecom Technology Journal, volume 6, no. 2, April 1988, G.D. Tattersall et al.: "Neural arrays for speech recognition", pages 140-163 see the whole document	1-18

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)

ANNEX TO THE INTERNATIONAL SEARCH REPORT
ON INTERNATIONAL PATENT APPLICATION NO.

GB 8800710
SA   24117

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report.
The members are as contained in the European Patent Office EDP file on 20/12/88
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A- 0085545	10-08-83	JP-A- 58129684 US-A- 4651289	02-08-83 17-03-87
WO-A- 8702805	07-05-87	JP-A- 62106590 EP-A- 0245508	18-05-87 19-11-87
GB-A- 1192554	20-05-70	None	